Detecting Rogue Switch and Device Behaviour Using Network Anomalies in LAN

Vijay Bhuse, Vijay Prathap Reddy Kanipakam, Rajvardhan Patil and Xinli Wang
Grand Valley State University, Allendale, MI, USA

bhusevij@gvsu.edu
kanipakv@mail.gvsu.edu
patilr@gvsu.edu
wangx@gvsu.edu

Abstract: Local Area Networks (LANs) are crucial for modern organizations, facilitating essential communication and data exchange in wired environments. However, wired LANs are susceptible to internal threats, exacerbated by "Bring Your Own Device" (BYOD) policies that increase vulnerability to rogue switches. These unauthorized switches, connected with just an Ethernet cable, can be installed by compromised employees or malicious insiders, undermining network security by intercepting and manipulating data traffic. These rogue switches, often plug-and-play devices, are particularly dangerous because they are difficult to detect and can be used to spy on network traffic or launch cyberattacks, further increasing organizational risks. This study presents a hybrid detection and mitigation framework that combines Dynamic ARP Inspection (DAI) with DHCP Snooping, Root Guard, and Port Security with Sticky MAC, alongside AI-driven anomaly detection. By integrating rule-based security mechanisms with supervised machine learning models, the system detects subtle deviations in network traffic and automates threat mitigation. This approach enhances detection accuracy, reduces false positives, and seamlessly integrates into existing security baselines. Experimental validation was conducted using GNS3-based lab simulations with a consistent network topology to evaluate detection effectiveness and dataset generation. Various Layer 2 attacks, including ARP spoofing, MAC flooding, and STP root bridge manipulation, were introduced to assess detection accuracy. The AI-enhanced system, trained with supervised learning using Logistic Regression, achieved 100% accuracy and an F1-score of 100% across all three attack scenarios, demonstrating its reliability in mitigating Layer 2 threats. The findings emphasise the effectiveness of combining AI-driven anomaly detection with traditional network security mechanisms to enhance LAN security. Unlike conventional reactive approaches, this framework enables proactive, real-time detection and mitigation, adapting to evolving threats and eliminating reliance on manual monitoring. The ability to detect subtle variations in network traffic behaviour ensures greater adaptability against sophisticated attacks. By continuously learning and refining detection models, the system provides scalable, intelligent, and future-ready network protection against increasingly advanced Layer 2 threats.

Keywords: LAN, Rogue switch and device, Dynamic ARP, Root guard, Port security, AI anomaly detection

1. Introduction

Local Area Networks (LANs) are crucial for enterprise communication and data exchange. However, their reliance on trusted internal infrastructure makes them vulnerable to unauthorised modifications, particularly through rogue switches and devices. Unlike external threats blocked by firewalls and intrusion detection systems, rogue switches blend into the network, allowing unauthorised access while remaining undetected. These unauthorised switches enable rogue devices to connect, bypass security controls, and exploit freely accessible or unmanaged switches that lack proper security enforcement, making them prime entry points for network intrusions. While ARP spoofing, MAC flooding, and root bridge manipulation can originate from compromised hosts, rogue switches integrate seamlessly into the network fabric, amplifying these threats while evading detection. Their persistence allows them to intercept traffic, launch attacks, or manipulate network configurations without immediately triggering security mechanisms. Some rogue switches even operate without a MAC address, making them especially difficult to detect in wired LAN environments.

Rogue entities passively observe or manipulate network traffic, evading conventional monitoring and making detection difficult in dynamic LAN environments where frequent changes occur. Traditional security mechanisms like Dynamic ARP Inspection (DAI), DHCP Snooping, Root Guard, and Port Security rely on static rules, limiting their adaptability to evolving threats. While effective against known attacks, they struggle to detect covert rogue switches, highlighting the need for a dynamic, real-time anomaly detection strategy.

Our unified framework enhances detection by integrating individual static security rules into a centralized system and building AI-driven anomaly detection on top of it. This unique approach analyzes anomaly patterns, employs foundational models, and identifies rogue switches and devices across multiple Layer 2 threats and evolving behaviours. To validate its effectiveness, we conducted extensive experiments to assess its ability to detect and mitigate rogue network elements in wired LAN environments.

The paper is structured as follows: Section 2 covers related work, Section 3 presents case studies, Section 4 discusses AI anomaly-based detection, and Section 5 concludes with key findings and future directions.

2. Related Work

Detecting rogue switches and devices in wired LANs remains a persistent challenge, largely due to their ability to passively blend into the network and evade traditional detection mechanisms. Prior studies (Bhuse et al., 2019; Quitiquit & Bhuse, 2022) have explored detection techniques such as MAC whitelisting, DHCP behaviour analysis, and port state monitoring. While effective at identifying unauthorised devices, these approaches do not provide a complete picture of how rogue switches can actively facilitate Layer 2 attacks, including ARP spoofing, MAC flooding, and STP manipulation, nor do they offer a comprehensive strategy to mitigate such threats.

While Layer 2 threats present serious security risks, most existing research continues to focus on host-based anomalies, leaving rogue switches an underexplored vector for long-term compromise. To address individual Layer 2 threats, several works have proposed focused anomaly detection strategies. Sun et al. (2022) applied unsupervised Autoencoder neural networks to detect ARP spoofing behaviour, showcasing the potential of deep learning for ARP-based anomalies. Similarly, SDN-based approaches have enabled centralized control and dynamic flow validation to mitigate spoofing attacks. Saritakumar et al. (2023) proposed an SDN-based ARP spoofing detection module using a RYU controller and Open vSwitch in an emulated Mininet environment. However, these methods are typically attack-specific and lack a unified, scalable framework capable of addressing multiple, concurrent Layer 2 threats in dynamic LAN environments.

In parallel, recent research has shown that supervised learning algorithms offer efficient and interpretable alternatives to deep learning systems. Kolukisa et al. (2024) proposed a Logistic Regression model optimised using a Parallel Artificial Bee Colony algorithm. Their approach achieved competitive results against deep learning models on benchmark datasets. This highlights the effectiveness of Logistic Regression for binary classification tasks. It also reinforces its suitability for structured anomaly detection in network environments.

Building on both rule-based methods and machine learning advances, research has progressively explored hybrid detection frameworks. In line with this direction, this paper provides a clear mapping to design robust LAN environments by leveraging static rule-based controls as a foundation. It further enables the seamless integration of artificial intelligence into existing LAN infrastructures, supporting both static and dynamic network conditions for effective rogue switch and device detection.

3. Case Studies

These attacks were identified as high-risk threats due to their frequent exploitation in real-world LAN breaches and their ability to persist undetected via rogue switches. By exploiting inherent LAN security gaps, they compromise traffic integrity, disrupt network stability, and weaken authentication mechanisms. This study evaluates these threats and demonstrates how the proposed framework effectively detects and mitigates them in a controlled lab environment. The experiments were conducted using GNS3 for network emulation, Cisco switches, Windows, Kali Linux, VPCS, Ettercap, Macof, and Wireshark for traffic capture and analysis.

3.1 Case Study on ARP Spoofing by Rogue Switches and Devices – Detection and Mitigation

Insider threats like ARP spoofing in wired LANs have heightened due to complex network architectures. Attackers use forged ARP messages for man-in-the-middle attacks and unauthorized data access, while rogue switches bypass traditional controls like ARP inspection, making detection difficult. This case study explores ARP spoofing in a simulated environment, focusing on a rogue device connected via a rogue switch, analysing the interception and manipulation of communication between legitimate devices. It assesses the effectiveness of detection methods in identifying rogue equipment and mitigating their network impact.

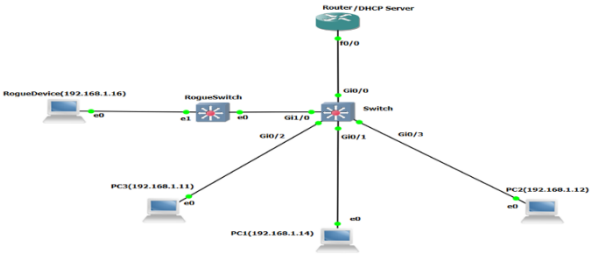
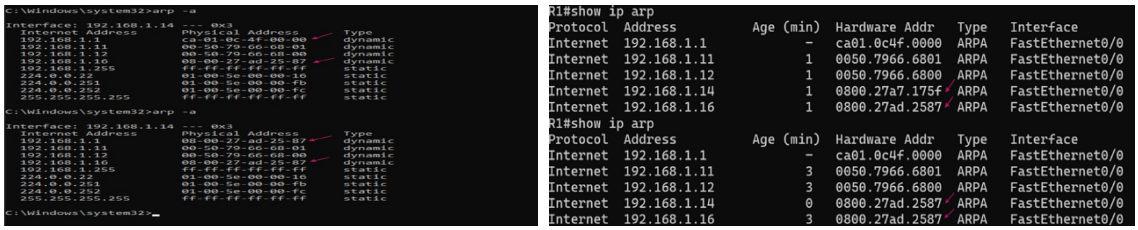


Figure 1.0: Topology Overview

In this network setup, a router (192.168.1.1) providing DHCP services connects through a legitimate switch to three PCs (PC1 at 192.168.1.14, PC2 at 192.168.1.12, PC3 at 192.168.1.11). A rogue switch connected to the legitimate one introduces unauthorized elements, with a rogue device (Kali Linux with IP 192.168.1.16) attached to it, using this connection to initiate an ARP spoofing attack using the Ettercap tool.

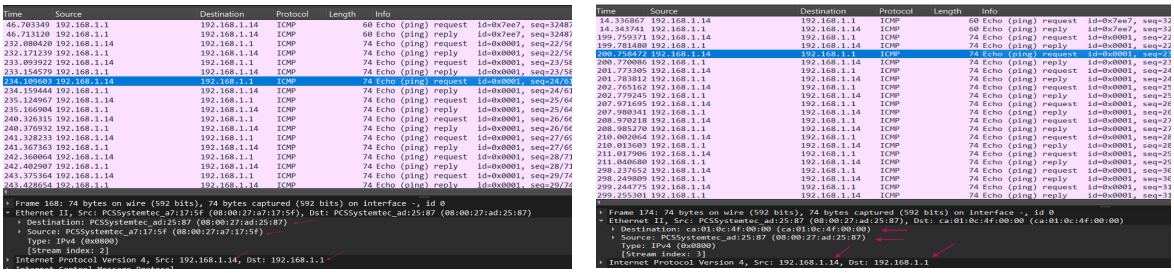


```
C:\Windows\system32>arp -a

Interface: 192.168.1.14 --- 0x3
  Internet Address      Physical Address      Type
  192.168.1.1           ca-01-0c-4f-00-00     dynamic
  192.168.1.11          00-50-79-66-68-01     dynamic
  192.168.1.12          00-50-79-66-68-00     dynamic
  192.168.1.16          08-00-27-ad-25-87     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Windows\system32>arp -a

Interface: 192.168.1.14 --- 0x3
  Internet Address      Physical Address      Type
  192.168.1.1           08-00-27-ad-25-87     dynamic
  192.168.1.11          00-50-79-66-68-01     dynamic
  192.168.1.12          00-50-79-66-68-00     dynamic
  192.168.1.16          08-00-27-ad-25-87     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Windows\system32>_
```

```
R1#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1             -   ca01.0c4f.0000  ARPA   FastEthernet0/0
Internet  192.168.1.11            1   0050.7966.6801  ARPA   FastEthernet0/0
Internet  192.168.1.12            1   0050.7966.6800  ARPA   FastEthernet0/0
Internet  192.168.1.14            1   0800.27a7.175f  ARPA   FastEthernet0/0
Internet  192.168.1.16            1   0800.27ad.2587  ARPA   FastEthernet0/0
R1#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1             -   ca01.0c4f.0000  ARPA   FastEthernet0/0
Internet  192.168.1.11            3   0050.7966.6801  ARPA   FastEthernet0/0
Internet  192.168.1.12            3   0050.7966.6800  ARPA   FastEthernet0/0
Internet  192.168.1.14            0   0800.27ad.2587  ARPA   FastEthernet0/0
Internet  192.168.1.16            3   0800.27ad.2587  ARPA   FastEthernet0/0
```

Figure 1.1 and 1.2: IP-MAC Inconsistency Observed in PC1's and Router ARP Cache Before and After ARP Spoofing

The ARP spoofing attack manipulates the ARP cache of PC1 and the router by sending ARP reply to messages (Opcode 2) without any ARP request, causing them to map each other's IP addresses to the MAC address of a rogue device (08:00:27:ad:25:87). This misdirection allows the attacker to intercept or alter communications between PC1 and the router by exploiting the automatic trust inherent in ARP responses.



```
Time      Source          Destination     Protocol  Length  Info
 46.703349 192.168.1.1    192.168.1.14    ICMP        60 Echo (ping) request  id=0x7ee7, seq=32487
 46.713120 192.168.1.14   192.168.1.1     ICMP        60 Echo (ping) reply    id=0x7ee7, seq=32487
232.080420 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=22/5632
232.171239 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=22/5632
233.093922 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=23/5888
233.154579 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=23/5888
234.109401 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=24/6144
234.159444 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=24/6144
235.124967 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=25/6400
235.166904 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=25/6400
240.326315 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=26/6656
240.376932 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=26/6656
241.328233 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=27/6912
241.367363 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=27/6912
242.360064 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=28/7168
242.402907 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=28/7168
243.375364 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=29/7424
243.428654 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=29/7424

▸ Frame 168: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface -, id 0
▾ Ethernet II, Src: PCSSystemtec_a7:17:5f (08:00:27:a7:17:5f), Dst: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87)
  ▸ Destination: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87)
  ▸ Source: PCSSystemtec_a7:17:5f (08:00:27:a7:17:5f)
    Type: IPv4 (0x0800)
    [Stream index: 2]
▸ Internet Protocol Version 4, Src: 192.168.1.14, Dst: 192.168.1.1
```

```
Time      Source          Destination     Protocol  Length  Info
 14.336867 192.168.1.14   192.168.1.1     ICMP        60 Echo (ping) request  id=0x7ee7, seq=32
 14.343741 192.168.1.1    192.168.1.14    ICMP        60 Echo (ping) reply    id=0x7ee7, seq=32
199.759371 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=22
199.781480 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=22
200.250472 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=23
200.770086 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=23
201.773305 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=24
201.783812 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=24
202.765162 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=25
202.779245 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=25
207.971695 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=26
207.980341 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=26
208.970218 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=27
208.985270 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=27
210.002064 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=28
210.013603 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=28
211.017906 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=29
211.040680 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=29
298.237652 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=30
298.249809 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=30
299.244775 192.168.1.14   192.168.1.1     ICMP        74 Echo (ping) request  id=0x0001, seq=31
299.255301 192.168.1.1    192.168.1.14    ICMP        74 Echo (ping) reply    id=0x0001, seq=31

▸ Frame 174: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface -, id 0
▾ Ethernet II, Src: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87), Dst: ca:01:0c:4f:00:00 (ca:01:0c:4f:00:00)
  ▸ Destination: ca:01:0c:4f:00:00 (ca:01:0c:4f:00:00)
  ▸ Source: PCSSystemtec_ad:25:87 (08:00:27:ad:25:87)
    Type: IPv4 (0x0800)
    [Stream index: 3]
▸ Internet Protocol Version 4, Src: 192.168.1.14, Dst: 192.168.1.1
```

Figure 1.3 and 1.4: Traffic flow between the legitimate switch and PC1 (192.168.1.14) and between Legitimate Switch and Router (192.168.1.1)

The figure 1.3 shows ARP spoofing attack manipulates the ARP tables of both PC1 and the router, redirecting PC1's traffic to a rogue device (MAC address 08:00:27:ad:25:87) instead of the router. As a result, the rogue device intercepts, monitors, or alters traffic before passing it to the router. Traffic analysis in figure 1.4 shows that packets from PC1 falsely appear to originate from the rogue device in the router's ARP cache, allowing the attacker to also intercept responses from the router, thereby controlling the communication flow between PC1 and the router.

3.2 Traffic Analysis (Post-DAI Implementation)

Switches do not validate ARP messages or check IP addresses, as they operate primarily on Layer 2 (Data Link Layer). If a rogue device sends spoofed ARP packets with a legitimate MAC address but a false IP address, the switch typically forwards these without detection. Dynamic ARP Inspection (DAI), leveraging the DHCP snooping binding table, enables switches to verify ARP packets against both IP and MAC addresses, effectively blocking and logging spoofed ARP requests. Despite ongoing spoofing attempts by a rogue device targeting both the router (192.168.1.1) and PC1 (192.168.1.14), the implementation of DAI successfully prevented these attacks, maintaining network integrity.

```
17:40:31.048: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.14/ca01.052f.0000/192.168.1.1/17:40:30 UTC Wed Oct 16 2024]
17:40:32.052: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.1/0800.27a7.175f/192.168.1.14/17:40:31 UTC Wed Oct 16 2024]
17:40:32.052: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.14/ca01.052f.0000/192.168.1.1/17:40:31 UTC Wed Oct 16 2024]
17:40:33.064: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.1/0800.27a7.175f/192.168.1.14/17:40:32 UTC Wed Oct 16 2024]
17:40:33.064: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.14/ca01.052f.0000/192.168.1.1/17:40:32 UTC Wed Oct 16 2024]
17:40:43.093: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.1/0800.27a7.175f/192.168.1.14/17:40:42 UTC Wed Oct 16 2024]
17:40:43.093: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.14/ca01.052f.0000/192.168.1.1/17:40:42 UTC Wed Oct 16 2024]
17:40:53.156: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.1/0800.27a7.175f/192.168.1.14/17:40:52 UTC Wed Oct 16 2024]
17:40:53.156: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Gi1/0, vlan 1.([0800.27ad.2587/192.168.1.14/ca01.052f.0000/192.168.1.1/17:40:52 UTC Wed Oct 16 2024]
```

Figure 1.5: Shows the Legitimate Switch Logs

Switch logs revealed that multiple ARP spoofing attempts were made from the interface (GigabitEthernet1/0) by a rogue device with MAC address 08:00:27:ad:25:87. These attempts aimed to spoof the IP addresses of the router (192.168.1.1) and a connected PC (192.168.1.14). Dynamic ARP Inspection (DAI) effectively blocked these attempts, as confirmed by logs recording denied ARP requests. This ensured the network remained secure, demonstrating DAI's effectiveness in detecting and preventing network compromises by rogue devices.

Summary: The detailed logging capabilities of Dynamic ARP Inspection (DAI) allowed network administrators to accurately trace and identify suspicious activities to GigabitEthernet1/0, pinpointing the source of unauthorized behaviour. This automated system reduces the need for manual monitoring by providing real-time detection and blocking of rogue switches and devices. DAI's effectiveness in swiftly detecting and mitigating threats ensures robust and continuous LAN security across both simple and complex network environments.

3.3 Case Study on Root Bridge Attacks by Rogue Switches – Detection, Mitigation, and Isolation

This case study explores the use of Root Guard within the Spanning Tree Protocol (STP) to detect, prevent, and isolate Root Bridge attacks. Root Bridge, central to network path calculations, is selected based on the lowest Bridge ID, considering priority and MAC address. Rogue switches can disrupt this by sending superior BPDUs with a lower priority, aiming to be chosen as the Root Bridge, which can lead to network instability, loops, and security risks. While BPDU Guard secures edge ports by disabling them when unauthorized BPDUs are detected, it is unsuitable for uplink ports. Root Guard addresses this by monitoring uplink ports and blocking any rogue switch attempting to influence the Root Bridge selection by placing these ports in a root-inconsistent state. This ensures the legitimate Root Bridge maintains control, preserving network stability and topology integrity. The case study shows Root Guard's effectiveness in maintaining a secure and stable STP environment.

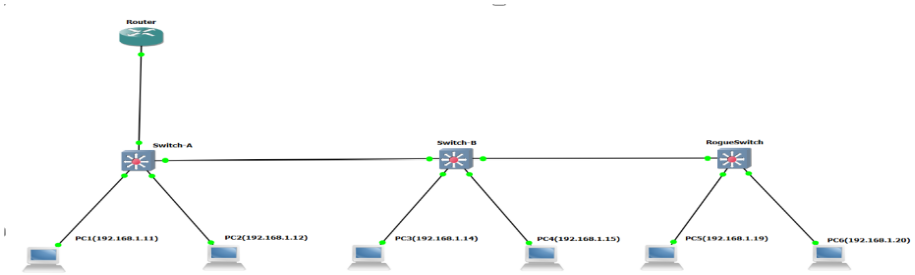


Figure 2.0: Topology Overview

The topology consists of a Router connected to Switch A, the legitimate Root Bridge, which links to Switch B, and a rogue Switch is connected to Switch B. Switch A connects to PC1 and PC2, while Switch B connects to PC3 and PC4. The rogue Switch is configured to disrupt the network by sending superior BPDUs, connecting PC5 and PC6.

```
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     0cca.d33b.0000
             Cost        4
             Port        1 (GigabitEthernet0/0)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28673  (priority 28672 sys-id-ext 1)
             Address     0c19.5a8a.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec
```

```
Spanning tree enabled protocol ieee
Root ID    Priority    1
           Address     0c2a.3630.0000
           Cost        8
           Port        2 (GigabitEthernet0/1)
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
           Address     0cca.d33b.0000
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time  300 sec
```

Figure 2.1 and 2.2: Switch B Spanning-Tree Information and Switch A Spanning-Tree Information

While configuring the wired LAN, Switch A is elected as the Root Bridge in the Spanning Tree Protocol (STP) setup. The output confirms that Switch A's Bridge Priority is (24577), while Switch B's priority is (28673), validating that the election process is consistent with the legitimate setup.

After the rogue switch joined the network with a default bridge priority of (32769), which was higher than Switch A's priority, it did not initially become the Root Bridge. However, after manually changing the priority of the rogue switch to 1 (the lowest possible value), it began sending superior BPDUs to the network. This led to the

rogue switch taking over the Root Bridge role, overriding Switch A. As a result, the network topology was disrupted, with the Root ID now reflecting the Bridge ID of the rogue switch, demonstrating how a rogue switch can manipulate STP to control the network.

3.4 Traffic Analysis (Post-Root Guard Implementation)

We implemented Root Guard on both Switch A and Switch B to ensure comprehensive protection in our smaller, controlled LAN environment. This setup isolates any rogue switch attempting to connect, preventing it from sending superior BPDUs and disrupting the Spanning Tree Protocol (STP) topology. While larger networks may apply Root Guard selectively on key uplink ports, our approach ensures network stability and simplifies management.

Figure 2.3 and 2.4: Switch B Logs

Figure 2.3 above indicates that after configuring Root Guard on Switch B, it was enabled on the uplink port (GigabitEthernet0/1) connecting to the rogue switch. Upon detecting these unauthorized BPDUs, Root Guard immediately blocked communication through this port and isolated the rogue switch.

The output above figure 2.4 indicates that the uplink port (GigabitEthernet0/1) was marked as root-inconsistent, meaning it was effectively blocked, preventing the rogue switch from taking over the Root Bridge role.

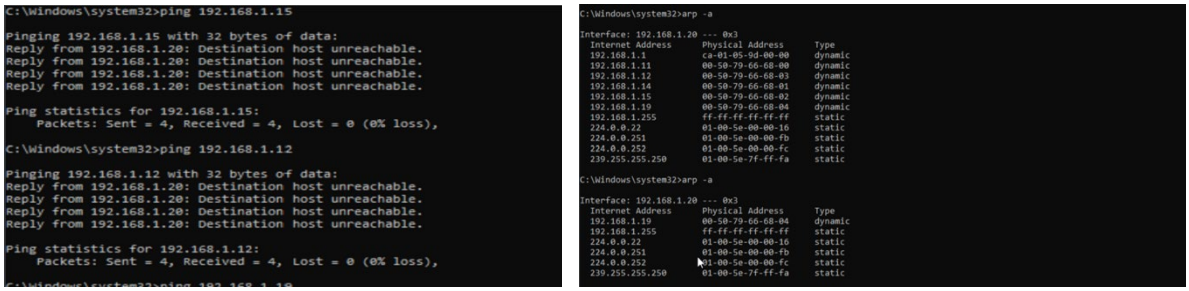


Figure 2.5 and 2.6: Ping test from PC6 and ARP Table of PC6

Before enabling Root Guard, the rogue switch could send superior BPDUs, allowing devices connected to it to communicate successfully with devices connected to other switches in the network. The "Destination host unreachable" errors, displayed in Figure 2.5, confirm the network disruption caused by the rogue switch. After enabling Root Guard on Switch B, these devices were fully isolated, and communication attempts to devices connected to Switch A, Switch B, or the router failed. The cleared ARP cache, as shown in Figure 2.6, further confirms the successful isolation of the rogue switch and Root Guard's role in preserving the legitimate STP topology.

Summary: This study demonstrated the effectiveness of Root Guard in mitigating Root Bridge attacks within a Local Area Network (LAN). By isolating rogue switches through root-inconsistent states and providing automated logs to identify compromised ports, Root Guard offers a practical method for network administrators to pinpoint and address rogue switches. The findings highlight Root Guard as a valuable solution for securing LAN environments against unauthorized manipulations of the Spanning Tree Protocol (STP).

3.5 Case Study on MAC Address Flooding by Rogue Switches and Devices – Detection and Mitigation and Isolating

The presence of multiple MAC addresses on a single access port signals a potential security threat, often caused by a rogue or unmanaged switch. In a secure LAN, each access port typically maps to a single device, but a rogue switch allows multiple unauthorized devices to transmit through one port, complicating traffic tracing and exposing the network to unauthorized access and malicious activity. To illustrate this, a MAC address flooding attack was conducted using a rogue switch to simulate multiple MAC addresses and overload the legitimate switch's MAC table. Once the table reached capacity, the switch defaulted to broadcasting packets across all ports, causing network congestion, disrupting traffic, and exposing sensitive data to interception.

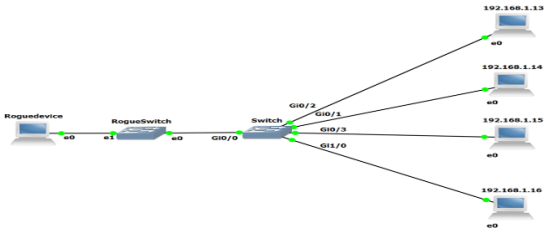
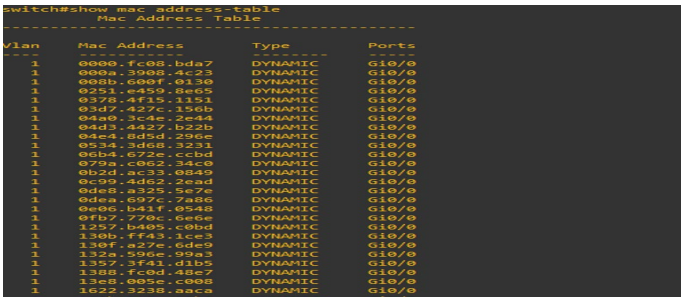


Figure 3.0: Topology Overview

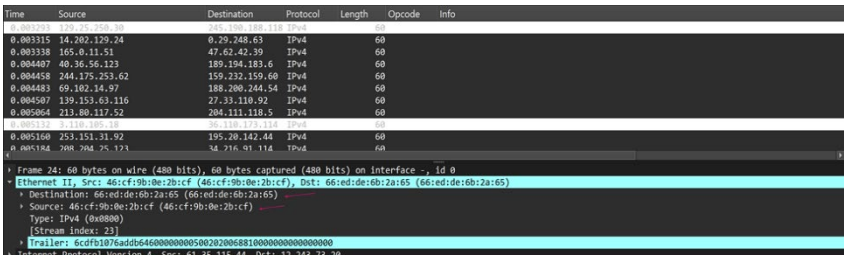
The topology includes four legitimate end devices (PCs with IP addresses 192.168.1.13 to 192.168.1.16) connected to separate ports on a core legitimate switch, which manages authorized devices and network traffic. On the other side, a rogue switch connects to port Gi0/0 of the legitimate switch and links to a rogue device (Kali Linux with IP 192.168.1.12). This rogue device generates packets with random MAC addresses, simulating multiple devices and triggering a MAC address flooding attack using the Macof tool. This attack forces the legitimate switch to store these fake addresses, overwhelming its capacity and compromising network traffic management.



```
switch#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0000.fc08.bda7    DYNAMIC     Gi0/0
   1    000a.3908.4c23    DYNAMIC     Gi0/0
   1    008b.600f.0130    DYNAMIC     Gi0/0
   1    0251.e459.8e65    DYNAMIC     Gi0/0
   1    0378.4f15.1151    DYNAMIC     Gi0/0
   1    03d7.427c.156b    DYNAMIC     Gi0/0
   1    04a0.3c4e.2e44    DYNAMIC     Gi0/0
   1    04d3.4427.b22b    DYNAMIC     Gi0/0
   1    04e4.8d5d.296e    DYNAMIC     Gi0/0
   1    0534.3d68.3231    DYNAMIC     Gi0/0
   1    06b4.672e.ccbd    DYNAMIC     Gi0/0
   1    079a.c062.34c0    DYNAMIC     Gi0/0
   1    0b2d.ac33.0849    DYNAMIC     Gi0/0
   1    0c99.4d62.2ead    DYNAMIC     Gi0/0
   1    0de8.a325.5e7e    DYNAMIC     Gi0/0
   1    0dea.697c.7a86    DYNAMIC     Gi0/0
   1    0e06.b41f.0548    DYNAMIC     Gi0/0
   1    0fb7.770c.6e6e    DYNAMIC     Gi0/0
   1    1257.b405.c0bd    DYNAMIC     Gi0/0
   1    130b.ff43.1ce3    DYNAMIC     Gi0/0
   1    130f.a27e.6de9    DYNAMIC     Gi0/0
   1    132a.596e.99a3    DYNAMIC     Gi0/0
   1    1357.3f41.d1b5    DYNAMIC     Gi0/0
   1    1388.fc0d.48e7    DYNAMIC     Gi0/0
   1    13e8.005e.c008    DYNAMIC     Gi0/0
   1    1622.3238.aaca    DYNAMIC     Gi0/0
```

Figure 3.1: Switch's MAC Address Table Observation

The MAC address table from the legitimate switch shows a flooding attack initiated by a rogue device via the rogue switch on port Gi0/0. Fake MAC addresses fill the table to capacity, preventing legitimate entries. Consequently, the switch broadcasts packets across all ports, causing network congestion, traffic disruption, and potential exposure of sensitive data. This overflow of MAC addresses on a single port highlights a critical vulnerability exploited by rogue switches, leading to network-wide disruptions and signalling their presence



```
Time      Source            Destination       Protocol  Length  Opcode  Info
0.003315  14.202.129.24     0.29.248.63       IPv4      60
0.003338  165.0.11.51       47.62.42.39       IPv4      60
0.004407  40.36.56.123      189.194.183.6     IPv4      60
0.004458  244.175.253.62    159.232.159.60    IPv4      60
0.004483  69.102.14.97      188.200.244.54    IPv4      60
0.004507  139.153.63.116    27.33.110.92      IPv4      60
0.005064  213.80.117.52     204.111.118.5     IPv4      60
0.005160  253.151.31.92     195.20.142.44     IPv4      60

Frame 24: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
Ethernet II, Src: 46:cf:9b:0e:2b:cf (46:cf:9b:0e:2b:cf), Dst: 66:ed:de:6b:2a:65 (66:ed:de:6b:2a:65)
  Destination: 66:ed:de:6b:2a:65 (66:ed:de:6b:2a:65)
  Source: 46:cf:9b:0e:2b:cf (46:cf:9b:0e:2b:cf)
  Type: IPv4 (0x0800)
  [Stream index: 23]
Trailer: 6cdfb1076addb64600000000050020200688100000000000000
```

Figure 3.2: Traffic Flow Captured Between Legit Switch and PC

Traffic analysis captures packets with spoofed MAC addresses, generated by the rogue device and forwarded through the rogue switch. The rogue switch functions as an entry point, facilitating the MAC address flooding attack by continuously introducing unique, fake addresses that overwhelm the legitimate switch's CAM (Content Addressable Memory) table. Once full, the switch indiscriminately broadcasts traffic across all ports.

3.6 Traffic Analysis (Post-Implementation of Port Security and Sticky MAC Addresses)

To mitigate the risk of MAC flooding and unauthorized access, port security is implemented to limit the number of MAC addresses each port can learn. If the number of connected devices exceeds this limit, the switch enforces measures such as shutting down the port or logging violations, effectively preventing unauthorized devices and halting attacks.

Sticky MAC addresses complement this by dynamically retaining the MAC addresses of trusted devices, ensuring only authorized devices can reconnect seamlessly. Together, these measures provide strong access control while maintaining network security and efficiency.

Figure 3.3: Sticky MAC and Figure 3.4: Port violation

Figure 3.3 shows the MAC address table of a legitimate switch with port security and Sticky MAC enabled. The MAC address 0800.27ad.2587 is associated with VLAN 1, confirming its presence within the same broadcast domain. The entry Labelled "Secure Sticky" indicates that the MAC address has been dynamically learned and bound to port Gi0/0. This configuration ensures only the device with this authorized MAC address can access the port, blocking unauthorized devices.

Figure 3.4 displays security violation logs triggered by Port Security, highlighting multiple unauthorized MAC addresses attempting to connect through port Gi0/0. Each violation reflects the rogue switch introducing new MAC addresses. After repeated violations, the switch transitions to an error-disable state, shutting down Gi0/0 to isolate and prevent further unauthorized access.

Figure 3.5: Port Security Info on Interface Gigabitethernet0/0

The port is in secure-shutdown mode due to 7,174 security violations caused by repeated attempts to exceed the allowed number of MAC addresses. A mismatch between the last detected source MAC address and the configured Sticky MAC address (0800.27ad.2587) confirms unauthorized activity, aiding administrators in identifying rogue devices.

Static vs. Dynamic LAN Environments: In static LANs, Sticky MAC addresses bind specific MAC addresses to ports, effectively "locking" them to authorized devices. Since devices rarely change, MAC aging is unnecessary, reducing congestion risks and ensuring stable access control. In dynamic LANs, MAC aging removes inactive MAC addresses after a set period (e.g., 300 seconds), preventing table congestion and allowing secure, flexible access for active devices.

Summary: This case study demonstrates how Port Security and Sticky MAC mitigate MAC flooding attacks in a LAN. Initially, the absence of MAC address restrictions allowed a rogue switch to introduce fake MAC entries, causing network congestion and disrupting communication. By enabling Port Security and Sticky MAC, the legitimate switch restricted unauthorized access, dynamically learned trusted MAC addresses, and detected rogue activity through security violations. These violations, logged automatically, provided administrators with critical information to identify and isolate the rogue switch and device. This approach effectively prevented further disruption, ensuring secure and stable LAN communication.

4. Enhancing Detection with AI-Based Anomaly Detection

Due to the lack of publicly available datasets relevant to Layer 2 threats, we generated our own to ensure accurate labelling, real-world relevance, and controlled testing conditions. The dataset was built by first capturing normal ('good') network traffic, then introducing a rogue switch and device to generate anomalous ('bad') traffic. This structured approach created a clear baseline and attack contrast, which was essential for training and evaluating our supervised learning models effectively within the context of the proposed framework.

AI Methodology and Algorithm: The AI integration introduces anomaly detection for case study scenarios using targeted datasets. Network traffic data was captured with Wireshark and pre-processed using Pandas. Logistic Regression was chosen for its computational efficiency and interpretability, making it suitable for real-time LAN security. Since the study focuses on specific rogue switch behaviours, a well-structured dataset enabled binary

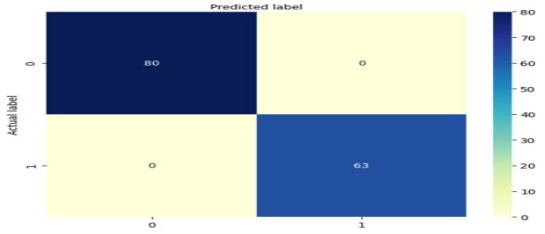
classification of rogue and non-rogue entities. To overcome dataset limitations, the model was optimized for smaller datasets, ensuring accurate detection without extensive training data.

The model uses the sigmoid activation function, which outputs probabilistic values between 0 and 1 for two-class classification. Data splitting was managed using Stratified Shuffle Split, ensuring balanced class distribution in the training and testing datasets, further enhancing the model's reliability and accuracy.

4.1 ARP Spoofing by Rogue Switches and Devices

The AI model detects these anomalies by analysing critical features such as ARP Opcode mismatches (MAC-to-IP inconsistencies), and abnormal reply to patterns. Duplicate IP usage across different MAC addresses further indicated spoofing attempts. These deviations from normal behaviour enabled accurate classification of rogue activity.

Dataset Overview: The dataset consisted of 713 rows (397 non-rogue, 316 rogue), with features such as *Request Time, Request Source, Request Destination, Opcode_1, Protocol, Reply Time, Reply Source, Reply Destination,* and *Opcode_2*. Labels: 0 (non-rogue) and 1 (rogue). It was divided into 570 training rows (80%) and 143 testing rows (20%), with a class distribution of 56% non-rogue and 44% rogue data points.



	precision	recall	f1-score	support
Without Rogue Switch	1.00	1.00	1.00	80
with Rogue Switch	1.00	1.00	1.00	63
accuracy			1.00	143
macro avg	1.00	1.00	1.00	143
weighted avg	1.00	1.00	1.00	143

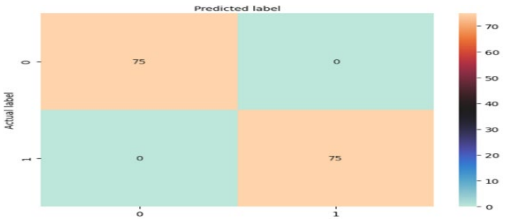
Figure 4.0 and 4.1: Confusion Matrix and Evaluation Metrics

The model detects IP-to-MAC inconsistencies, continuous Opcode 2 (reply) messages without corresponding Opcode 1 (request) messages, irregular timing of ARP replies from specific sources, and duplicate IP address usage. These real-time detections provide actionable insights, enabling effective mitigation of rogue network activity.

4.2 Root Bridge Manipulation by Rogue Switches

To detect anomalies in the Spanning Tree Protocol (STP), traffic was captured near the legitimate root bridge switch to ensure precise monitoring of topology-related events. The AI model analyses superior BPDU activity to identify unauthorized BPDUs with lower priority attempting to override the root bridge, root bridge MAC address changes to detect discrepancies, and path cost deviations signaling topology manipulation. Path cost, representing the cumulative cost of traversing network links to the root bridge, is closely monitored. An increase in path cost from 0 (indicating proximity to the legitimate root bridge) to a higher value near the root switch suggests potential interference by rogue switches. By focusing on traffic captured near the root bridge, the model effectively detects and flags rogue switch activities in real-time, preserving STP topology integrity.

Dataset Overview: The dataset comprised 749 rows (374 non-rogue, 375 rogue), with features like Root Priority, Root MAC Address, Path Cost, and Protocol. Labels: 0 (non-rogue), 1 (rogue). It was split into 599 training rows (80%) and 150 testing rows (20%), maintaining a balanced 50:50 class distribution.



	precision	recall	f1-score	support
Without Rogue Switch	1.00	1.00	1.00	75
with Rogue Switch	1.00	1.00	1.00	75
accuracy			1.00	150
macro avg	1.00	1.00	1.00	150
weighted avg	1.00	1.00	1.00	150

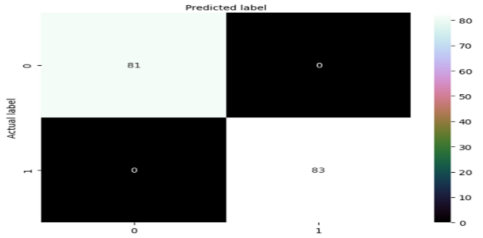
Figure 4.2 and 4.3: Confusion Matrix and Evaluation Metrics

These results validate the AI's ability to identify deviations in root bridge behaviour effectively. By continuously monitoring superior BPDUs, root MAC changes, and path cost variations, the model provides real-time insights into rogue switch activity, ensuring the integrity of the STP topology.

4.3 MAC Flooding by Rogue Switches and Devices

In the baseline case studies, MAC flooding overwhelms a switch's CAM table with fake MAC addresses, causing traffic broadcasts and congestion. The AI model enhances detection by analysing real-time data from Wireshark and correlating MAC addresses with legitimate switch ports. By monitoring MAC variability, traffic patterns, and CAM table saturation, it identifies rogue devices and enables real-time isolation, minimizing disruptions.

Dataset Overview: The dataset comprised 819 rows (404 non-rogue, 415 rogue), with features such as Request Time, Source IP, Destination IP, Protocol, Length, Source MAC, Destination MAC, Source Switch Port MAC, and Destination Switch Port MAC. Labels: 0 (non-rogue) and 1 (rogue). It was split into 656 training rows (80%) and 164 testing rows (20%), with a class distribution of 51% non-rogue and 49% rogue data points.



	precision	recall	f1-score	support
Without Rogue Switch	1.00	1.00	1.00	81
with Rogue Switch	1.00	1.00	1.00	83
accuracy			1.00	164
macro avg	1.00	1.00	1.00	164
weighted avg	1.00	1.00	1.00	164

Figure 4.4 and 4.5: Confusion Matrix and Evaluation Metrics

These metrics validate the AI's ability to detect MAC flooding attacks with precision. By integrating Wireshark data for granular analysis and correlating MAC addresses to specific switch ports, the model provides actionable insights in real time. This enables administrators to quickly isolate rogue switches and devices, minimizing disruptions and securing network integrity.

5. Conclusion

This paper presents a framework for detecting rogue switches and devices in wired LANs by combining traditional methods with AI-driven anomaly detection. Techniques such as Dynamic ARP Inspection (DAI), Root Guard, and Port Security effectively address threats like ARP spoofing, MAC flooding, and root bridge manipulation, as demonstrated in case studies. However, traditional static rule-based methods rely on predefined patterns, making them less effective against evolving threats. To enhance detection capabilities, the proposed framework integrates AI to analyse real-time deviations in network activity. The Logistic Regression model, selected for its binary classification efficiency, demonstrated high accuracy, achieving 100% in specific scenarios using datasets generated through GNS3 simulations that mimic realistic network environments. By combining static rule-based methods with AI-driven approaches, the framework provides robust, scalable, and adaptive network security. Future work will involve testing the model with larger, real-world datasets and incorporating diverse traffic patterns to further validate its effectiveness and detect emerging rogue behaviours.

References

Bhuse, V., Kalafut, A. and Dohn, L., 2019. Detection of a Rogue Switch in a Local Area Network. In *The Fourteenth International Conference on Internet Monitoring and Protection (ICIMP)*.

Bhuse, V., Vellaboina, Y. and Wang, X., 2024, June. Enhancing Network Security: Rogue Switch Detection and Prevention in Local Area Network. In *European Conference on Cyber Warfare and Security* (Vol. 23, No. 1, pp. 66-73).

Cisco Systems, 2007. Understanding and Configuring Dynamic ARP Inspection. [online] Available at: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/dynarp.html [Accessed 23 September 2024].

Cisco Systems, 2018. Enhance Spanning Tree Protocol (STP) with Root Guard. [online] Available at: https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/10588-74.html [Accessed: 13 October 2024].

Cisco Systems, 2020. Port Security. *Security Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9300 Switches)*. [online] Available at: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/17-3/configuration_guide/sec/b_173_sec_9300_cg/port_security.html [Accessed 18 November 2024].

GNS3, 2024. *GNS3 Documentation*. [online] Available at: https://docs.gns3.com/docs/ [Accessed 14 October 2024].

Kali.org, n.d. *macof: Flood a Switched LAN with Random MAC Addresses*. [online] Available at: https://www.kali.org/tools/dsniff/#macof [Accessed: 25 October 2024].

Kolukisa, B., Dedeturk, B.K., Hacilar, H. and Gungor, V.C. (2024) 'An efficient network intrusion detection approach based on logistic regression model and parallel artificial bee colony algorithm', *Computer Standards & Interfaces*, 89, p.103808.

Prins, K. and Bhuse, V., 2018, June. Forced vacation: A rogue switch detection technique. In *European Conference on Cyber Warfare and Security* (pp. 390-399). Academic Conferences International Limited.

Quitiqut, T. and Bhuse, V., 2022, March. Utilizing Switch Port Link State to Detect Rogue Switches. In *International Conference on Cyber Warfare and Security* (Vol. 17, No. 1, pp. 272-278).

Saritakumar, N., Anusuya, K.V. and Krishnakumar, S., 2023, February. Detection of ARP Spoofing Attacks in Software Defined Networks. In *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)* (pp. 422-426). IEEE.

Sun, Y., Ochiai, H. and Esaki, H., 2022, January. Suspicious ARP Activity Detection and Clustering Based on Autoencoder Neural Networks. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 743-744). IEEE.

Taknev, A., 2024. Beginner's Guide to Ettercap. *HackerCool Magazine*. [online] Available at: https://www.hackercoolmagazine.com/beginners-guide-to-ettercap/ [Accessed 16 October 2024].

Vyncke, E. and Paggen, C., 2008. Attacking the Spanning Tree Protocol. In: *LAN Switch Security: What Hackers Know About Your Switches*. Indianapolis: Cisco Press.

Wireshark, 2024. *Wireshark User's Guide*. [online] Available at: https://www.wireshark.org/docs/wsug_html_chunked/ [Accessed 20 October 2024].